**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ**

**РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Кубанский Государственный Университет**

Кафедра информационных технологий

ОТЧЕТ

о выполнении лабораторной работы №5

по дисциплине «Коллективная разработка приложений»

Выполнили:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ст. гр. 44 Владарчук Д.В.\_\_\_\_

ст. гр. 44 Емельяненко А.А.\_\_

ст. гр. 44 Иванов А.А.\_\_\_\_\_\_\_

Проверил:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

доц. каф. ИТ Полетайкин А.Н.

Краснодар

2025

**Тема: Разработка и анализ приложения**

Цель: получить практические навыки в управлении этапом разработки кода программных компонентов.

**Задание**

1. Перевести задачи командного проекта в состояние **In Progress**.

2. Создать в решении проект для разработки приложения ИС.

3. Провести необходимые работы по этапу разработки кода приложения ИС в составе командного проекта.

4. Провести анализ кода приложения.

**Ход работы**

**1. Задачи командного проекта**

Наша команда перевела все задачи в статус **In Progress** через Azure DevOps Server (Рисунок 1):

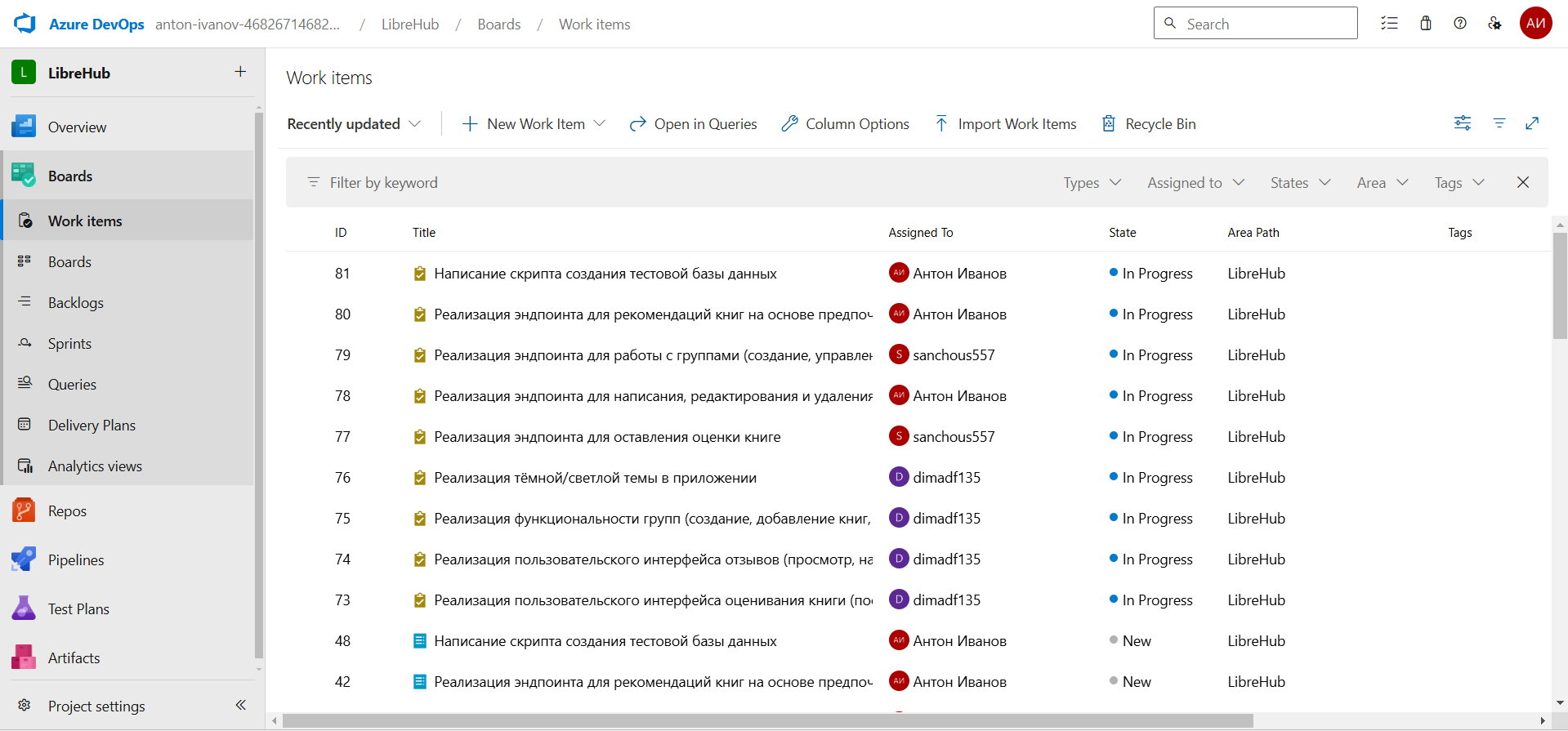


Рисунок 1 – задачи в статусе In Progress

Теперь каждый член команды при подключении к проекту в Microsoft Visual Studio на вкладке «Рабочие элементы» может увидеть задачи, которые назначены ему на выполнение (Рисунок 2):

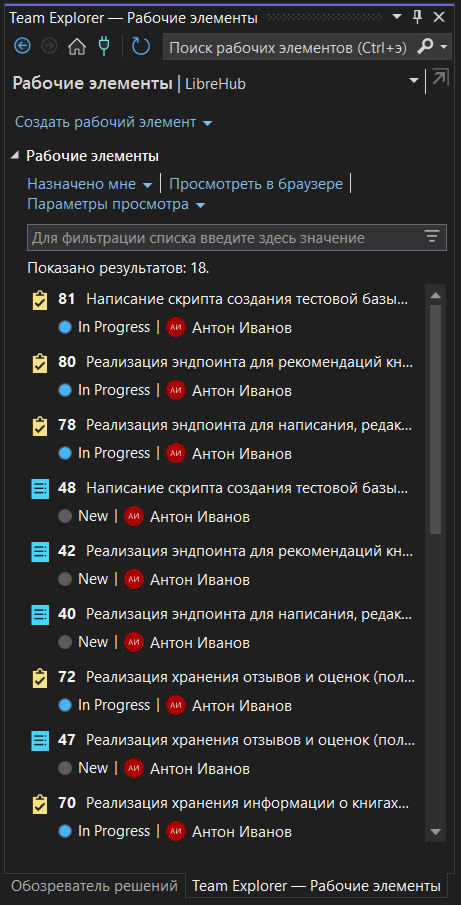


Рисунок 2 – личные задачи сотрудника

Однако внутри Microsoft Visual Studio нет раздела «Моя работа», а следовательно, и отсутствуют контекстные меню с окнами для работы с задачами непосредственно в Visual Studio, вместо этого ссылки перенаправляют на веб-портал проекта (Azure DevOps Server), где можно редактировать задачи (Рисунок 3):

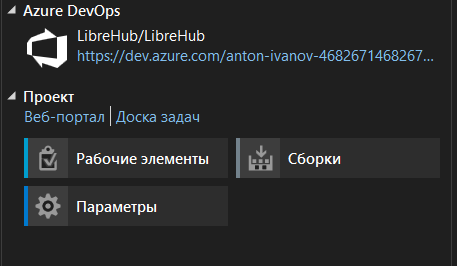


Рисунок 3 – отсутствие вкладки «Моя работа»

**2. Создание проекта для разработки приложения ИС**

Наша команда приняла решение реализовать требуемый продукт, разработав клиентское мобильное приложение на Kotlin в среде Android Studio (Рисунок 4), а серверную часть – на C# в виде Web-API приложения на основе ASP.NET с использованием native AOT (Рисунок 5):

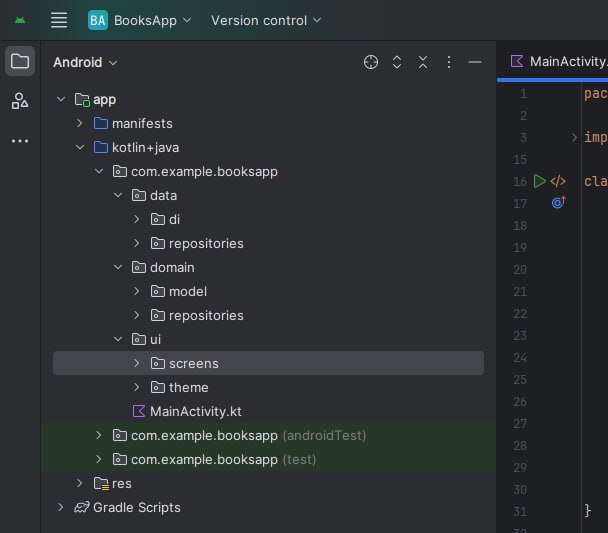


Рисунок 4 – шаблон создаваемого мобильного приложения

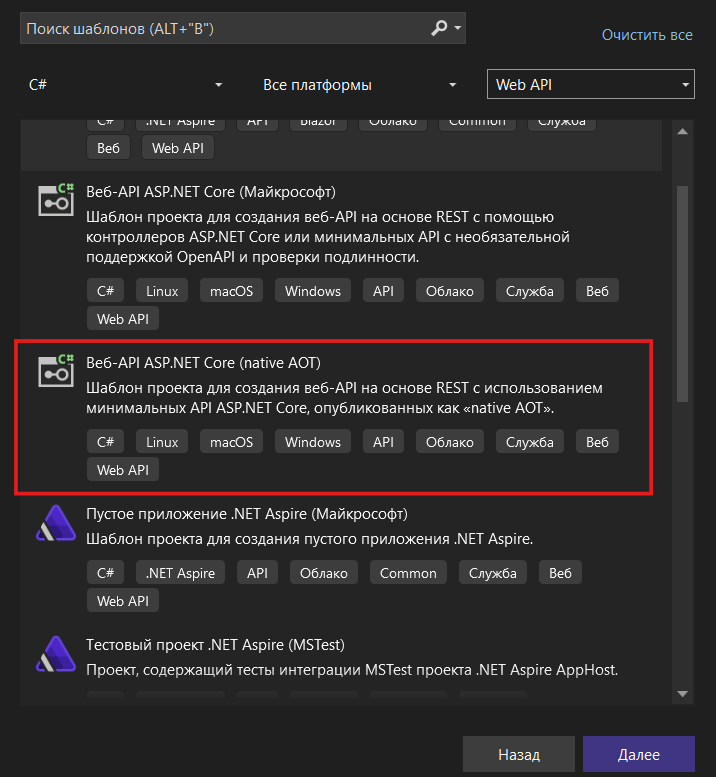


Рисунок 5 – шаблон создаваемого серверного приложения

В результате проект в Microsoft Visual Studio имеет такую структуру (Рисунок 6):

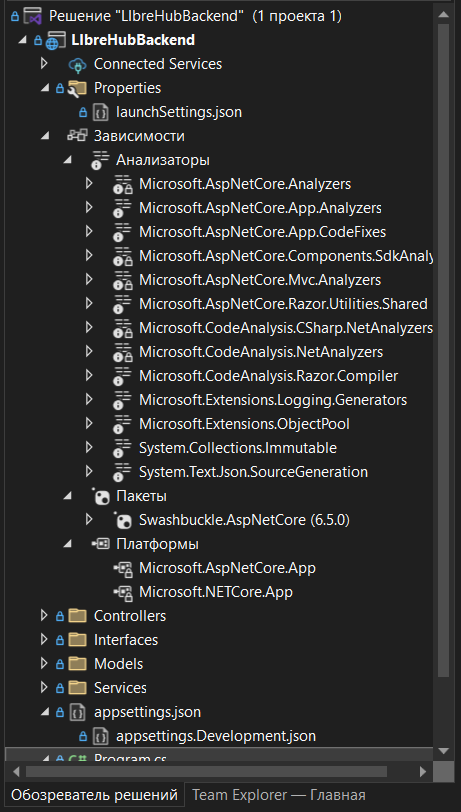


Рисунок 6 – структура шаблона приложения

Далее для удобной локальной разработки каждого участника команды был создан репозиторий проекта (Рисунок 7), который каждый участник команды клонировал и вёл локальную разработку посредством коммитов (Рисунок 8):

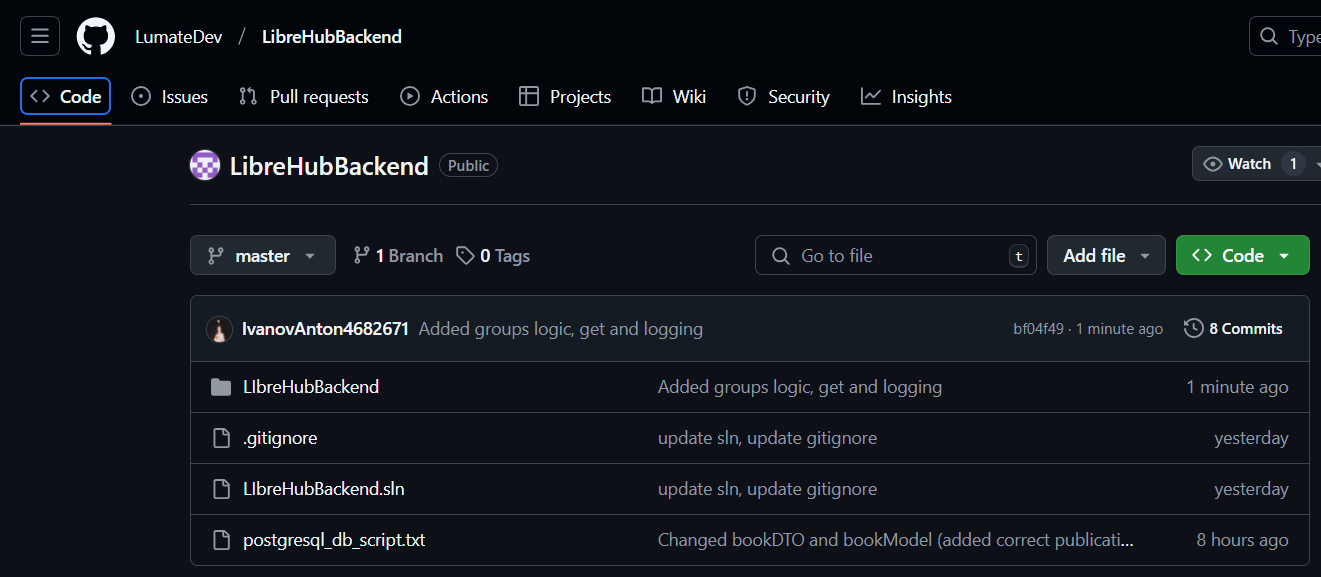


Рисунок 7 – репозиторий GitHub, в котором ведётся разработка проекта

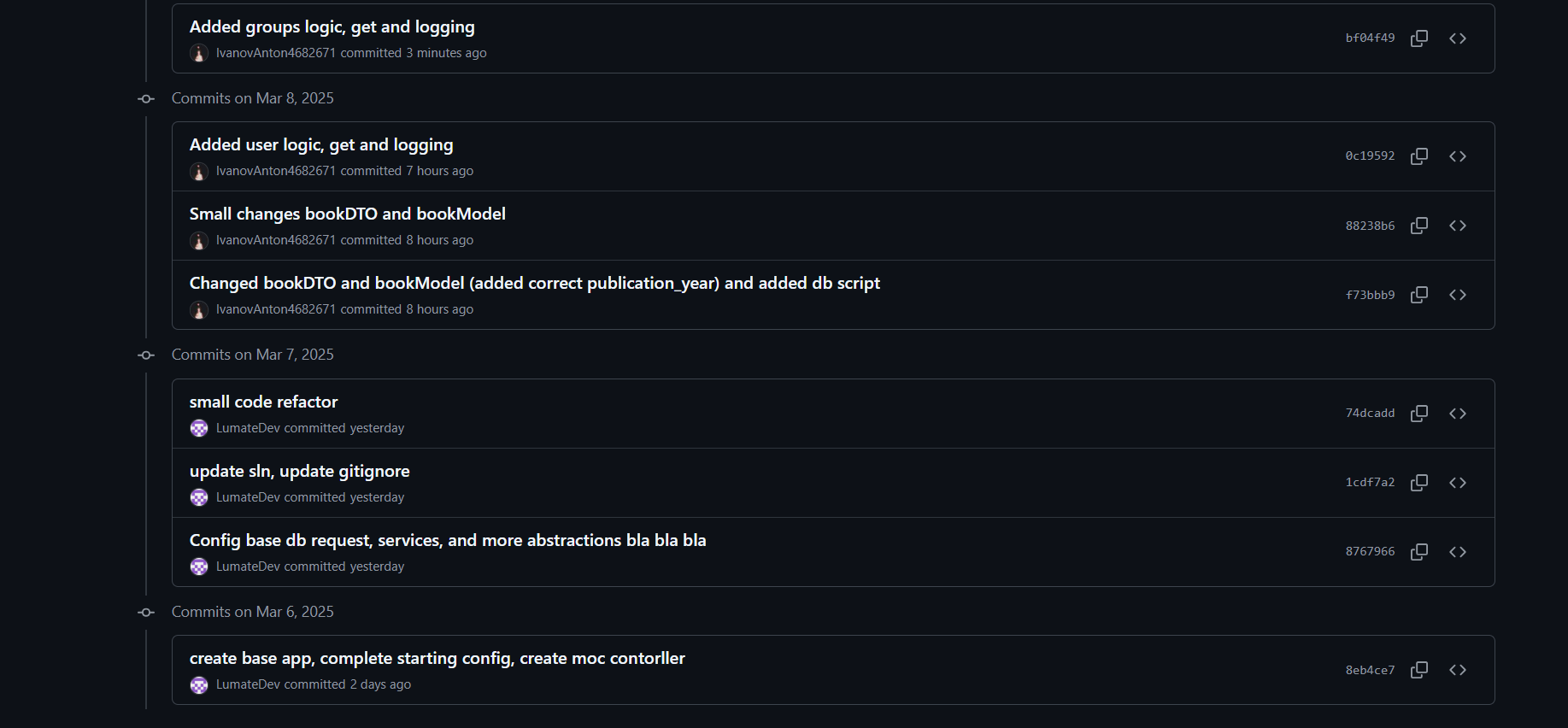


Рисунок 8 – коммиты при разработке проекта членами команды

**3. Разработка кода приложения ИС**

Как уже упоминалось выше, участники команды ведут разработку посредством коммитов (реализуется какая-то задача по плану спринта, после чего это изменение проекта сохраняется и отправляется в общий репозиторий).

При разработке приложения на данном этапе возникли трудности с базой данных, а именно с установлением корректного соединения при переходе с .NET 6 на .NET 8, из-за чего было принято решений перейти на базу данных PostgreSQL.

На текущем этапе приложение корректно устанавливает соединение с базой данных и имеет корректно работающие точки входа (эндпоинты) для получения записей из базы данных (Рисунки 9-10):

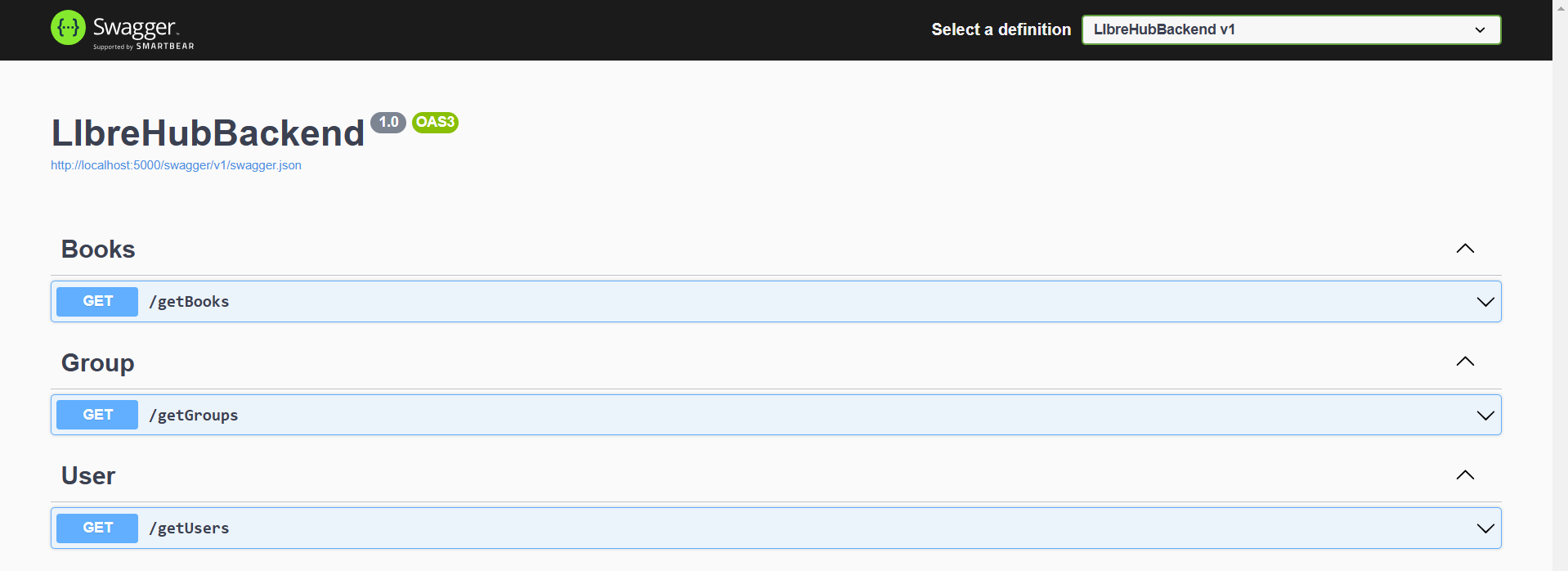


Рисунок 9 – экран тестирования эндпоинтов проекта



Рисунок 10 – тестирование конкретного эндпоинта (получение списка книг)

**4. Анализ кода приложения**

Наша команда провела анализ кода (как было предложено в методичке) (Рисунок 11):

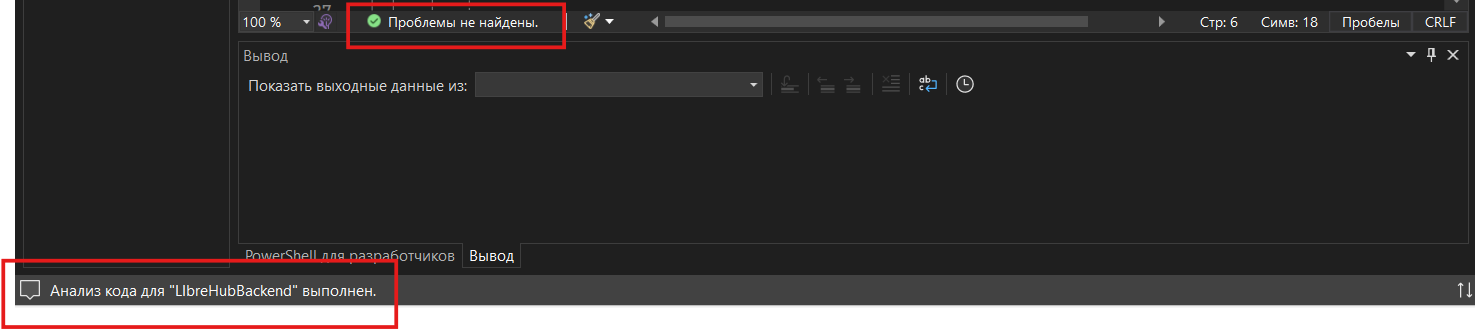


Рисунок 11 – результат анализа кода проекта

В предыдущих версиях проекта данный анализ выдавал некоторые предупреждения в методах, которые мы использовали синхронно (без использования асинхронного вызова await, что снижало скорость работы приложения) (Рисунок 12), но сейчас мы это исправили (как и показано на рисунке 11):

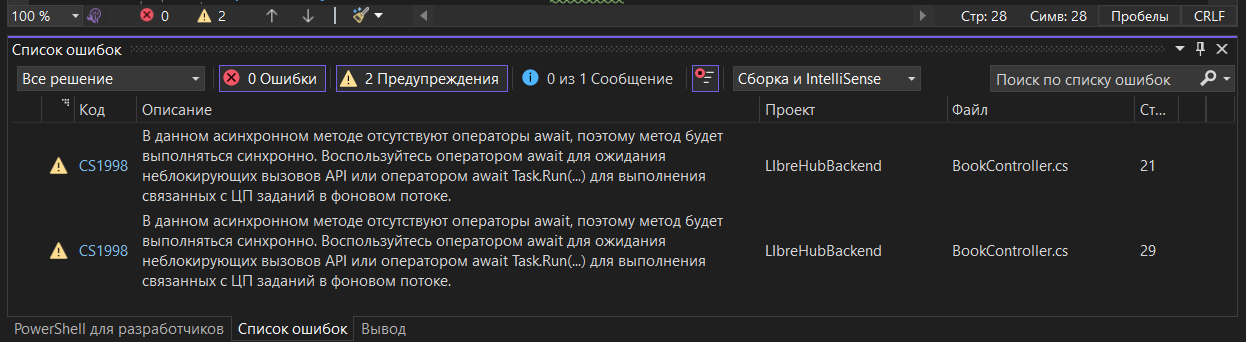


Рисунок 12 – предыдущие запуски анализа кода

Также мы провели оценку метрик кода, как предлагалось в методичке. Результаты метрик кода представлены на рисунке 13:

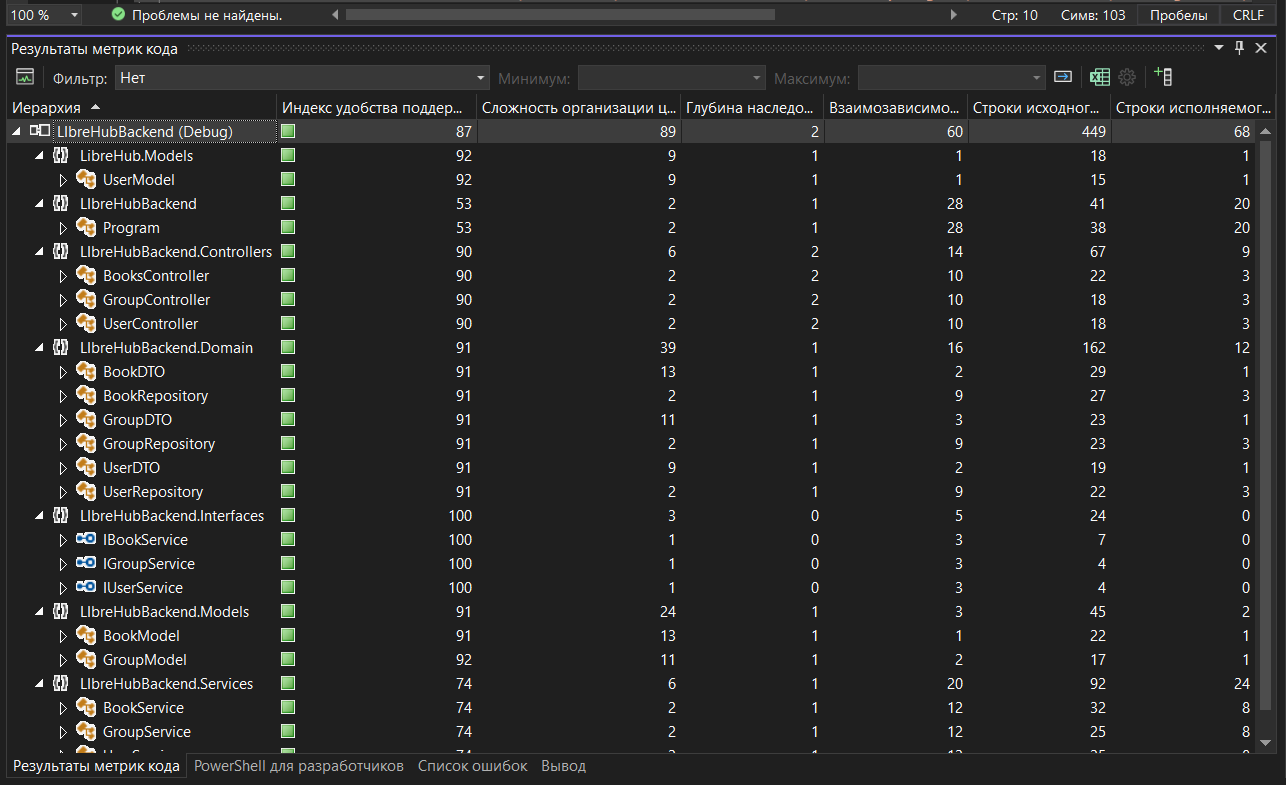


Рисунок 13 – метрики кода приложения

Пояснения: для первого столбца предпочтительны большие значения, а для остальных – меньшие.

**Вывод**

Вывод: в ходе выполнения данной работы была проделана работа над кодом разрабатываемого приложения. Также все текущие задачи были переведены в состояние In Progress. Были созданы проекты по готовым шаблонам, был проведён анализ написанного кода, получены метрики.